

IN THE CLAIMS

1. (original) A computer-implemented method for software error recovery, comprising:
- compiling program source code into a first set of object code with a first compiler;
 - compiling the program source code into a second set of object code with a second compiler;
 - identifying checkpoints in the first and second sets of object code, each checkpoint in the first set of object code corresponding to a checkpoint in the second set of object code;
 - associating sets of data objects with the checkpoints;
 - automatically generating executable checkpoint code for execution at the checkpoints, the checkpoint code configured to store state information of the associated data objects for recovery if execution of the program is interrupted;
 - executing the first set of object code;
 - storing the state information in executing the checkpoint code; and
 - upon detecting an error in execution of the first set of object code, resuming execution of the program using the second set of object code.
2. (original) The method of claim 1, further comprising:
- upon detecting an error in execution of the first set of object code, initially re-executing the first set of object code; and
 - resuming execution using the second set of object code if the first set of object code fails in re-execution.
3. (original) The method of claim 2, further comprising re-executing the first set of object code a selected number of times before resuming execution using the second set of object code.
4. (original) The method of claim 3, further comprising ceasing resumption of execution of the first and second sets of object code if an error is detected in executing both sets of object code.
5. (original) A computer-implemented method for software error recovery, comprising:
- compiling program source code into a first set of object code with a first compiler;

compiling the program source code into a second set of object code with a second compiler;

identifying checkpoints in the first and second sets of object code, each checkpoint in the first set of object code corresponding to a checkpoint in the second set of object code;

associating sets of data objects with the checkpoints;

automatically generating executable checkpoint code for execution at the checkpoints, the checkpoint code configured to store state information of the associated data objects for recovery if execution of the program is interrupted;

executing the first set of object code;

storing the state information in executing the checkpoint code; and

upon detecting an error in execution of the first set of object code, selecting between the first set of object code and the second set of object code in resuming execution of the program.

6. (original) The method of claim 5, further comprising:

upon detecting an error in execution of the first set of object code, initially re-executing the first set of object code; and

resuming execution using the second set of object code if the first set of object code fails in re-execution.

7. (original) The method of claim 6, further comprising re-executing the first set of object code a selected number of times before resuming execution using the second set of object code.

8. (original) The method of claim 7, further comprising ceasing resumption of execution of the first and second sets of object code if an error is detected in executing both sets of object code.

9. (original) An apparatus for software error recovery, comprising:

means for compiling program source code into a first set of object code with a first compiler;

means for compiling the program source code into a second set of object code with a second compiler;

means for identifying checkpoints in the first and second sets of object code, each checkpoint in the first set of object code corresponding to a checkpoint in the second set of object code;

means for associating sets of data objects with the checkpoints; and

means for automatically generating executable checkpoint code for execution at the checkpoints, the checkpoint code configured to store state information of the associated data objects for recovery if execution of the program is interrupted;

means for executing the first set of object code;

means for storing the state information in executing the checkpoint code; and

means for resuming execution of the program using the second set of object code upon detecting an error in execution of the first set of object code.

10. (currently amended) An apparatus ~~computer-implemented method~~ for software error recovery, comprising:

a1 means for compiling program source code into a first set of object code with a first compiler;

means for compiling the program source code into a second set of object code with a second compiler;

means for identifying checkpoints in the first and second sets of object code, each checkpoint in the first set of object code corresponding to a checkpoint in the second set of object code;

means for associating sets of data objects with the checkpoints;

means for automatically generating executable checkpoint code for execution at the checkpoints, the checkpoint code configured to store state information of the associated data objects for recovery if execution of the program is interrupted;

means for executing the first set of object code;

means for storing the state information in executing the checkpoint code; and

selecting between the first set of object code and the second set of object code in resuming execution of the program upon detecting an error in execution of the first set of object code.

11. (original) A computer program product configured for causing a computer to perform the steps comprising:

compiling program source code into a first set of object code with a first compiler;
compiling the program source code into a second set of object code with a second compiler;

identifying checkpoints in the first and second sets of object code, each checkpoint in the first set of object code corresponding to a checkpoint in the second set of object code;

associating sets of data objects with the checkpoints;

automatically generating executable checkpoint code for execution at the checkpoints, the checkpoint code configured to store state information of the associated data objects for recovery if execution of the program is interrupted;

executing the first set of object code;

storing the state information in executing the checkpoint code; and

upon detecting an error in execution of the first set of object code, selecting between the first set of object code and the second set of object code in resuming execution of the program.

12. (new) The computer program product of claim 11, further configured for causing a computer to perform the steps comprising:

upon detecting an error in execution of the first set of object code, initially re-executing the first set of object code; and

resuming execution using the second set of object code if the first set of object code fails in re-execution.

13. (new) The computer program product of claim 12, further configured for causing a computer to perform the step comprising re-executing the first set of object code a selected number of times before resuming execution using the second set of object code.

14. (new) The computer program product of claim 13, further configured for causing a computer to perform the step comprising ceasing resumption of execution of the first and second sets of object code if an error is detected in executing both sets of object code.
